

Attorney Docket No.: 03824.P003

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

TITLE OF THE INVENTION:

METHOD AND APPARATUS FOR AUTOMATIC FORM FILLING

INVENTORS:

MARK E. PENNELL
ANTHONY MARTIN

Prepared by

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(503) 684-6200

684-6200

This application claims the benefit of U.S. Provisional Application No. 60/107,791, filed November 10, 1998.

BACKGROUND OF THE INVENTION

5 Field of the Invention

The present invention relates to the field graphical user interfaces and more particularly to a method for entering information into a form on a screen display associated with an electronic device.

10 Description of the Related Art

Many a site on the Internet's World Wide Web (hereafter "web site". "web page" or simply "site") require the entry of various information in order to gain full access to the site and the services offered by the site. For example, many commercial sites require a user to set up an account and, in doing so, to provide various levels of personal information. Typically, the information is relatively repetitive from site to site—e.g., name, address, telephone number, 15 electronic mail (email) address, credit card number, etc. In some cases, the information must be entered each time the user attempts to use the site. In other cases, an account is actually set up for the user and maintained -- the user needs only to enter the full information the first time the site is accessed.

20 An example of an account set up screen is provided in Figure 1. The figure illustrates a screen shot 100 of a web site accessed via, for example, web browser software executing on computing device such as a personal computer. The web site provides for online ordering, in this particular instance, of cookies. As can be seen, the user is invited to complete the shipping and

billing information by visiting each data field and entering the appropriate information (e.g., name 101, address 102, phone number 103, email address 104, etc.). Alternatively, at some web sites, the user may click on a button, hyperlink, etc., to log in and fill in ordering information automatically (presuming the user has a previously set up account.)

5 One method of addressing the inconvenience of repetitive data entry of account information is the so-called "wallet" technology. Using "wallets", a user may enter certain information (name, address, billing/credit card information) once and sites that run the particular wallet technology will be able to receive the information without requiring the user to reenter the data. Unfortunately, this technology requires sites to execute the wallet technology in order to allow a user to benefit from it. A diagram 200 illustrating a particular embodiment of the wallet technology is shown in Figure 2.

One other method of addressing the inconvenience of repetitive data entry is the so-called "type-ahead" technology in which the user's computing system attempts to "remember" certain information and, if a user starts to type a sequence of characters using, for example, a keyboard or other character input device, the system attempts to recognize the character sequence and complete the sequence. For example, if the user named John Smith starts to type his name in a name field, the system may recognize the user is typing "John Smith" after the user has only typed "Joh" and automatically fill in the remaining "n Smith".

Unfortunately, the type ahead technology is limited in that it may or may not correctly recognize the phrase being typed and implementations are typically browser software dependent. Moreover, the type ahead technology requires the user to independently visit each field in a form, rather than filling in multiple fields with a single click.

Thus, it would be useful to provide a method and apparatus which reduces the inconvenience of repetitive data entry. It would be particularly useful to provide a method and apparatus which was not browser dependent and which did not require implementation by each individual web site in order to allow a user to benefit from it.

66004 " 6600460

(iii) \mathcal{H}_1 and \mathcal{H}_2 are Hilbert spaces
 (iv) \mathcal{H}_1 and \mathcal{H}_2 are separable
 (v) \mathcal{H}_1 and \mathcal{H}_2 are reflexive
 (vi) \mathcal{H}_1 and \mathcal{H}_2 are Banach spaces
 (vii) \mathcal{H}_1 and \mathcal{H}_2 are normed spaces
 (viii) \mathcal{H}_1 and \mathcal{H}_2 are topological vector spaces
 (ix) \mathcal{H}_1 and \mathcal{H}_2 are linear spaces
 (x) \mathcal{H}_1 and \mathcal{H}_2 are vector spaces
 (xi) \mathcal{H}_1 and \mathcal{H}_2 are modules
 (xii) \mathcal{H}_1 and \mathcal{H}_2 are algebras
 (xiii) \mathcal{H}_1 and \mathcal{H}_2 are rings
 (xiv) \mathcal{H}_1 and \mathcal{H}_2 are groups
 (xv) \mathcal{H}_1 and \mathcal{H}_2 are monoids
 (xvi) \mathcal{H}_1 and \mathcal{H}_2 are semigroups
 (xvii) \mathcal{H}_1 and \mathcal{H}_2 are lattices
 (xviii) \mathcal{H}_1 and \mathcal{H}_2 are posets
 (xix) \mathcal{H}_1 and \mathcal{H}_2 are preorders
 (xx) \mathcal{H}_1 and \mathcal{H}_2 are partial orders
 (xxi) \mathcal{H}_1 and \mathcal{H}_2 are total orders
 (xxii) \mathcal{H}_1 and \mathcal{H}_2 are linear orders
 (xxiii) \mathcal{H}_1 and \mathcal{H}_2 are well-orders
 (xxiv) \mathcal{H}_1 and \mathcal{H}_2 are ordinals
 (xxv) \mathcal{H}_1 and \mathcal{H}_2 are cardinals
 (xxvi) \mathcal{H}_1 and \mathcal{H}_2 are alephs
 (xxvii) \mathcal{H}_1 and \mathcal{H}_2 are beths
 (xxviii) \mathcal{H}_1 and \mathcal{H}_2 are cofinalitys
 (xxix) \mathcal{H}_1 and \mathcal{H}_2 are regularitys
 (xxx) \mathcal{H}_1 and \mathcal{H}_2 are strong regularitys
 (xxxi) \mathcal{H}_1 and \mathcal{H}_2 are weak regularitys
 (xxxii) \mathcal{H}_1 and \mathcal{H}_2 are normalitys
 (xxxiii) \mathcal{H}_1 and \mathcal{H}_2 are countabilitys
 (xxxiv) \mathcal{H}_1 and \mathcal{H}_2 are separabilitys
 (xxxv) \mathcal{H}_1 and \mathcal{H}_2 are second countabilitys
 (xxxvi) \mathcal{H}_1 and \mathcal{H}_2 are first countabilitys
 (xxxvii) \mathcal{H}_1 and \mathcal{H}_2 are Lindelöfnesss
 (xxxviii) \mathcal{H}_1 and \mathcal{H}_2 are compactnesss
 (xxxix) \mathcal{H}_1 and \mathcal{H}_2 are sequential compactnesss
 (xl) \mathcal{H}_1 and \mathcal{H}_2 are countable compactnesss
 (xli) \mathcal{H}_1 and \mathcal{H}_2 are pseudocompactnesss
 (xlii) \mathcal{H}_1 and \mathcal{H}_2 are realcompactnesss
 (xliii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} -closednesss
 (xliv) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} -normalitys
 (xlv) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} -paracompactnesss
 (xlvi) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} -metacompactnesss
 (xlvii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} -collectionwise normalitys
 (xlviii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} -screenabilitys
 (xlvix) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π -screenabilitys
 (l) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} -normalitys
 (li) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} -paracompactnesss
 (lii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} -metacompactnesss
 (liiii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} -collectionwise normalitys
 (liv) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} -screenabilitys
 (lv) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π -screenabilitys
 (lvi) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -normalitys
 (lvii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -paracompactnesss
 (lviii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -metacompactnesss
 (lvix) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -collectionwise normalitys
 (lvi) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -screenabilitys
 (lvii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π -screenabilitys
 (lviii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -normalitys
 (lix) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -paracompactnesss
 (lxi) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -metacompactnesss
 (lxii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -collectionwise normalitys
 (lxiii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -screenabilitys
 (lxiv) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π -screenabilitys
 (lxv) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -normalitys
 (lxvi) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -paracompactnesss
 (lxvii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -metacompactnesss
 (lxviii) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -collectionwise normalitys
 (lxix) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} -screenabilitys
 (lxx) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π - \mathcal{C} - π -screenabilitys
 (lxxi) \mathcal{H}_1 and \mathcal{H}_2 are \mathcal{C} - π - \mathcal{C} - π - $\$

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

Fig. 1 is an exemplary form displayed by a web browser.

Fig. 2 illustrated a prior art wallet technology.

Fig. 3 illustrates a network as may utilize an embodiment of the invention.

Fig. 4 illustrates a form helper window as may be utilized by an embodiment of the present invention.

Fig. 5 illustrates a form helper window as may be utilized by an embodiment of the present invention.

Fig. 6 illustrates a form helper window as may be utilized by an embodiment of the present invention.

Fig. 7 illustrates a form helper window as may be utilized by an embodiment of the present invention.

Fig. 8 illustrates a login helper window as may be utilized by an embodiment of the present invention.

Fig. 9 illustrates a login helper window as may be utilized by an embodiment of the present invention.

For ease of reference, reference numerals in the accompanying drawings typically are in the form "drawing number" followed by two digits, xx; for example, reference numerals may be numbered 3xx. In certain cases, a reference numeral may be introduced on one drawing and the same reference numeral may be utilized on other drawings to refer to the same item.

DETAILED DESCRIPTION OF THE INVENTION

Figure 3 provides a diagram illustrating an overall system implementing an embodiment of the present invention. In the described embodiment, a user computing device, such as user computer 301, is automated with browser automation software 302. The browser automation software interfaces with any of a number of web browsers 303 such as Netscape Navigator available from Netscape Corporation of Mountain View, California or Internet Explorer available from Microsoft Corporation of Redmond, Washington. As a user moves between web pages in the World Wide Web using browser 303, the browser automation program 302 communicates with the browser and determines the Universal Resource Locator (URL) of the web site 306 being browsed. In certain embodiments, functionality of the browser automation program 302 may be added to the browser program 303 rather than executing the automation program 302 as a separate executable program.

In the described embodiment, the browser automation program 302 may gain knowledge of the format of a form encountered on any number of web sites. For purposes of this invention, a web site for which the format of the form has been learned by the browser automation program 302 is termed a "scripted" site. One method for the browser automation program to gain this knowledge is for the user to have previously filled out the same form. The browser automation program 302 then associates the content and order of the fields for the form with the content of personal data in the user database 304 (e.g., the program 302 learns that the field named "Name" on a particular form should be associated with the user's name in the user database 304.)

A second method for the browser automation program to gain this knowledge is for the form to have been analyzed and information stored regarding the fields and expected contents. This may be done, for example, for popular or well known web sites that utilize forms. The

information may be stored locally on each user's computer 301 or may be stored at a central location accessible to the user via network 307, such as the browser automation home site 305.

In an embodiment that stores this information at the home site 305, when a new URL is encountered, the home site 305 is contacted over the network 307. (It should be noted that the network could be the Internet or an intranet). In certain embodiments, information may be stored on the user's computer allowing local identification of which forms are stored at the home site 305. For example, a hash code may be developed to allow local (at the user's computer) determination of whether the form is scripted, i.e., whether information regarding the format of the form is stored, at the home site.

In addition, when encountering a form, whether for the first or a subsequent time, the browser automation program 302 may analyze the underlying structure of the form to determine if there are fields for which data is available from the user database 304. Typically, this process may involve analyzing the HyperText Markup Language (HTML), eXtensible Markup Language (XML), or other underlying code received from the visited web site 306.

Regardless of the method, if a script is available for the form, a pop up dialog window 401 is displayed in conjunction with the visited web site. An example in Figure 4 shows pop up window 401 overlaying a portion of a visited web site, for which a partial screen shot 100 is illustrated. The pop up window 401 allows the user to automatically place the information displayed in the fields of the pop up window into the corresponding fields of the form provided at the web page that is displayed on the screen of the user's computer. The user may supply all of the listed information in pop up window 401 or may modify some or all of it before supplying it to the form. Fig. 7 illustrates a web page form 100 filled in automatically by selecting the "fill in" button.

The pop up window 401 is better viewed with reference to Fig. 5. The user may supply the necessary information for the form provided at the scripted site by selecting the "fill in" button 402 of pop up window 401 (assuming a script exists for the form or alternatively the program 302 can gain sufficient knowledge of the form from analyzing the underlying HTML).

5 The "fill in" button may be selected, for example, by performing a single click of a user input device such as a mouse. Alternatively, if the browser automation program 302 is unfamiliar with the form, the user is provided with the pop up window 601 shown in Fig. 6. (Fig. 6 does not illustrate the form for which information displayed in pop up window 601 may be supplied). Pop up window 601 generally is utilized the first time a form is encountered, so that the user may
10 select each of the individual fields in the window. The user may double click on any one particular field in pop up window 601 to supply only that field of information to the form. That information is supplied, in particular, to the currently focused field in the form displayed by the browser. The browser automation software then causes the browser to automatically advance the focus to the next field in the form, in a step wise fashion. For example, if the user double clicks
15 on the name field 602, only the name field is supplied from the pop up window to the name field 101 in the form displayed on screen 100. Alternatively, the user may "drag and drop" the contents of a field in pop up window 601 to the corresponding field in the form displayed on screen 100.

It should be further noted that the pop up windows illustrated in Figs 5 and 6 provide for
20 multiuser support. For example, if multiple individuals share the same computer or web browser software, information about each user may be stored and subsequently accessed by supplying a uniquely identifying user name as input to the browser automation program 302. A user can select their data by specifying their name at field 403 in pop up window 401. In one

embodiment, field 403 is set up as a pull down list providing for the ability to select one of multiple users or to add a new user.

Moreover, for each user, any one of a number of profiles 404 may be provided from which to select to fill in the form. For example, the user may click on different profiles for home, work, or other. The multiple profiles allow for different sets of data to be input into the form, e.g., shipping address, phone, fax, and email address. For instance, depending on whether the user desires to communicate with the provider of the web site from home, work, or some other logical or physical designation, the user can select a desired profile to provide the appropriate information necessary for the web site provider to communicate or transact with the user accordingly. Fig. 4 illustrates the user's home profile is selected.

Yet further flexibility is provided by the browser automation program in utilizing pull down lists 405 for many of the fields of personal information accessible via pop up window 401. The user may specify one of multiple shipping addresses, phone numbers, email addresses, etc, for each profile. Thus, if a user maintains multiple offices and wishes to register or otherwise communicate personal contact information to a particular web site, the user may specify one particular office address. The user may then register at another web site using a different office address, by selecting a different office address via the pull down list associated with the shipping address field in the pop up window 401.

The information displayed in the pop up window 401 may have been initially supplied directly by the user or may have been learned as the user entered data in the normal course of filling out forms on web pages. The data is stored, typically in an encrypted format, on the user's computer 301 as user data in database 304. When the browser automation program 302 is executed, the user is asked for a password in order to access the encrypted data. In one

embodiment, the data is stored in a separate file which may be copied by the user and transported from computer to computer. In one embodiment, the data within the file 304 is retained with time stamp information. Using the timestamp information, the browser automation program 302 may merge two user data files, keeping the most recent information from both files.

5 In addition to assisting in completing relatively long forms as was shown in Figs, 4-7, the browser automation program 302 can assist with other types of forms 800. An example is provided in Figure 8 in which a "login helper" pop up dialog window 801 is displayed overlaying login screen 800. For ease of reference, dialog window 801 is shown separately in Fig. 9 as well. The browser automation program 302 has learned the user's login names and
10 passwords for given web sites (in this case, the Microsoft msn Hotmail web site). One problem increasingly facing web users is the need to remember not only many passwords but also many user identifications, or "member names". The browser automation program stores in the user data file 304 the login member names and passwords (in an encrypted format) for sites for which the user has registered. When the user accesses the URL for a site, the user is presented with the
15 login helper 801. Login helper 801 allows the user to select the appropriate member name and automatically then enters the correct password for the user. Of course, the user may have multiple member names for a particular site and the browser automation programs 302 store each of the various member names. The user may select the desired member name from a pull down list 802 in pop up window 801. Note also that, as in the case of pop up window 401, login helper
20 window 801 provides for multiuser support, by allowing a user to select from one of multiple users via pull down list 803.

The user may be provided with the option of having the form filled in by selecting the login button 804, for example, via a single click of a mouse pointer device. Thus, for example,

ALTERNATIVES TO THE PREFERRED EMBODIMENT OF THE PRESENT INVENTION

There are, of course, alternatives to the described embodiment which are within the reach of one of ordinary skill in the relevant art. The present invention is intended to be limited only by the claims presented below.

5

Thus, what has been disclosed is a method and apparatus for entry of form data in a web browser.

03824.P003